Theoretical Computer Science 439 (2012) 69–82

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Almost optimal distributed M2M multicasting in wireless mesh networks[☆]

Qin Xin[a,b,*], Fredrik Manne[c], Yan Zhang[d], Xin Wang[e]

[a] *Faculty of Science and Technology, University of the Faroe Islands, Torshavn, Faroe Islands*
[b] *Department of Computer Science, University of Copenhagen, Copenhagen, Denmark*
[c] *Department of Informatics, The University of Bergen, Bergen, Norway*
[d] *Simula Research Laboratory, Oslo, Norway*
[e] *Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA*

A R T I C L E   I N F O

Article history:
Received 22 June 2011
Received in revised form 3 January 2012
Accepted 10 March 2012
Communicated by G. Ausiello

Keywords:
Broadcasting
Distributed algorithms
Gossiping
Multicast
Unit disk graphs
Wireless mesh networks

A B S T R A C T

Wireless Mesh Networking (WMN) is an emerging communication paradigm to enable resilient, cost-efficient and reliable services for the future-generation wireless networks. In this paper, we study the problem of multipoint-to-multipoint (M2M) multicasting in a WMN which aims to use the minimum number of time slots to exchange messages among a group of k mesh nodes in a multi-hop WMN with n mesh nodes. We study the M2M multicasting problem in a distributed environment where each participant only knows that there are k participants and it does not know who are other $k-1$ participants among n mesh nodes. It is known that the computation of an optimal M2M multicasting schedule is *NP-hard*. We present a fully distributed deterministic algorithm for such an M2M multicasting problem and analyze its time complexity. We show that if the maximum hop distance between any two out of the k participants is d, then the studied M2M multicasting problem can be solved in time $O(d\log^2 n + \frac{k\log^3 n}{\log k})$ with a polynomial-time computation, which is an almost optimal scheme due to the lower bound $\Omega(d + \frac{k\log n}{\log k})$ given by Chlebus et al. (2009) [5]. Our algorithm also improves the currently best known result with running time $O(d\log^2 n + k\log^4 n)$ by Gąsieniec et al. (2006) [13]. In this paper, we also propose a distributed deterministic algorithm which accomplishes the M2M multicasting in time $O(d+k)$ with a polynomial-time computation in unit disk graphs. This is an asymptotically optimal algorithm in the sense that there exists a WMN topology, e.g., a line, a ring, a star or a complete graph, in which the M2M multicasting cannot be completed in less than $\Omega(d+k)$ units of time.

1. Introduction

The Wireless Mesh Network (WMN) is a highly promising network architecture to converge the future-generation wireless networks. A WMN has the dynamic self-organization, self-configuration and self-healing characteristics, and additionally inherent flexibility, scalability and reliability advantages. In a WMN, the mesh nodes can communicate with

☆ A preliminary version of this paper appears in the Proceedings of the Sixth IEEE International Conference on Mobile Ad-hoc and Sensor Systems, IEEE MASS 2009.

* Correspondence to: Faculty of Science and Technology, University of the Faroe Islands, Noatun 3, FO 100, Torshavn, Faroe Islands. Tel.: +298 352575.
E-mail addresses: qinx@setur.fo (Q. Xin), fredikm@ii.uib.no (F. Manne), yanzhang@simula.no (Y. Zhang), xwang@ece.sunysb.edu (X. Wang).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.03.015

each other via multi-hop routing or forwarding [1]. There are two types of WMNs with respect to the mobility of the mesh nodes, i.e. fixed mesh nodes and mobile mesh nodes. The IEEE 802.11s mesh networks in Wireless Local Area Networks (Wireless LAN) is a kind of WMNs with fixed mesh nodes, where the Access Points (APs) can communicate with each other via multi-hop routing. Another example can be the WMN constructed by the mesh routers with fixed topology. If the mesh nodes are installed in different moving objects, e.g. buses, trains or aircrafts, the network can be the type of WMNs with mobile mesh nodes. In this paper, we focus on the WMNs with fixed mesh nodes.

Next generation WMNs are expected to support group communication applications such as distance learning, video conference, disaster recovery, distributed collaborative computing and so on. In such applications, any of the mesh nodes of a well-defined group may be required to send messages to all other mesh nodes in the group. The problem of exchanging messages within a fixed group of mesh nodes in the multi-hop WMNs is called *M2M (multipoint-to-multipoint) multicasting*.

Broadcasting and *gossiping* are two classical problems of information dissemination in WMNs. Broadcasting problem is to distribute a message from a distinguished *source* mesh node to all other mesh nodes in the WMN. Gossiping problem is to distribute all messages m_v initially holding by each mesh node v to all other mesh nodes in the WMN. In both problems, one of the main efficiency criteria is the time needed to complete the given communication task. M2M multicasting is a natural generalization of gossiping, in which the information exchange concerns not all mesh nodes of the WMN but only a subset of all mesh nodes, called *participants*.

Although either the algorithms for broadcasting or the algorithms for gossiping could be used to solve M2M multicasting problem, the former often does not scale well while the latter may not be efficient because an application may involve only a small fraction of the total number of mesh nodes of the underlying WMN.

In this paper we address the problem of minimizing the communication time of M2M multicasting in multi-hop WMNs. We assume that the network topology is known to all mesh nodes in the WMN. This assumption is not exceptional since the WMN with fixed mesh nodes are considered in this work. The exemplary network can be Wireless LAN mesh networks or mesh router based WMNs. For the M2M multicasting problem with k participants of a distance at most d hops between any pair of them, we assume that each participant only knows that there are k participants and the value of d. However, it does not know which $k-1$ out of $n-1$ wireless mesh nodes are other participants. Note also that the M2M multicasting problem has been shown to be NP-hard. Therefore, the design of an approximation algorithm in a deterministic and distributed fashion with nearly optimal performance in terms of time efficiency becomes extremely challenging.

The algorithms proposed for the M2M multicasting problem in this paper are deterministic distributed communication algorithms. The proposed deterministic communication algorithms can assure that the communication task will be completed successfully as long as no topology changes occur in the WMN during the execution of our algorithms. Another interesting aspect of deterministic communication in WMNs with fixed topologies is its close relation with randomized communication in ad-hoc WMNs or the WMNs with mobile mesh nodes. Note also that our main goal is the design of time efficient communication procedures. However it would not be difficult to increase the level of fault-tolerance in our algorithm at the expense of some extra time consumption.

Our contributions are summarized as follows:

- We show that if the maximum distance between any two out of k participants is d hops then M2M multicasting problem can be solved in time $O(d\log^2 n + \frac{k\log^3 n}{\log k})$ with a polynomial-time computation, which is an almost optimal scheme due to the lower bound $\Omega(d + \frac{k\log n}{\log k})$ by Chlebus et al. in [5]. Our algorithm also improves the currently best known result with running time $O(d\log^2 n + k\log^4 n)$, by Gąsieniec et al. in [13].
- In this paper, we also show that the M2M multicasting problem can be solved in time $O(d+k)$ with a polynomial-time computation in the unit disk graphs. This is an asymptotically optimal algorithm in the sense that there exists a WMN topology, e.g., a line, a ring, a star or a complete graph, in which the M2M multicasting cannot be accomplished in less than $\Omega(d+k)$ units of time.

It is worth to remarking that we also show that our M2M multicasting algorithms can be extended to the scenario when the maximum hop distance d between the participants is not known in advance with asymptotically similar running time which is based on the standard doubling methodology.

The rest of the paper is organized as follows. We introduce the related work in Section 2. The preliminary results are mentioned in Section 3. We present the almost optimal algorithm for M2M multicasting in general graphs in Section 4. The asymptotically optimal algorithm for M2M multicasting in unit disk graphs is presented in Section 5. We conclude the paper in Section 7.

2. Related work

The complexity of various communication problems in wireless networks is highly related to the particular setting and model parameters, and may change significantly depending on whether the nodes know the network topology or not, what communication models are assumed, and so on. According to the graph models used for wireless networks, communication problems in wireless networks can be divided into two main subareas, one dealing with general graphs and the other concerning unit disk graphs. In what follows, we introduce the related work of broadcasting, gossiping, and M2M multicasting in both general graphs and unit disk graphs.

Broadcasting in general graphs: Gaber and Mansour [14] showed that the broadcasting task can be completed in time $O(D + \log^5 n)$ for every n-vertex wireless radio network of diameter D. In [4], Chlamtac and Weinstein proved that the broadcasting task can be completed in time $O(D \log^2 n)$. An $\Omega(\log^2 n)$ time lower bound was proved for the family of graphs of radius 2 by Alon et al. [2]. In [12], Elkin and Kortsarz gave an efficient deterministic construction of a broadcasting schedule of length $D + O(\log^4 n)$. Recently, Gąsieniec et al. [17] showed that a $D + O(\log^3 n)$ schedule exists for the broadcast task, that works in *any* wireless radio network. In the same paper, the authors also provided an optimal randomized broadcasting schedule of length $D + O(\log^2 n)$. Very recently, a $O(D + \log^2 n)$-time deterministic broadcasting schedule for any wireless radio network was proposed by Kowalski and Pelc in [19]. This is asymptotically optimal unless $NP \subseteq BPTIME(n^{\mathcal{O}(\log\log n)})$ [19]. Nonetheless, for large D, in [9], a $D + O(\frac{\log^3 n}{\log\log n})$ time broadcasting scheme outperformed the one in [19], because of the larger coefficient of the D term hidden in the asymptotic notation describing the time evaluation of this latter scheme. Efficient broadcasting algorithms for several special types of network topologies can be found in [11,23]. For general wireless networks, however, it is known that the computation of an optimal broadcasting schedule is NP-hard, even if the underlying graph is embedded in the plane [3,24].

Broadcasting in unit disk graphs: In [10], Dessmark and Pelc presented a broadcasting schedule of length at most $2400D$. In [15], Gandhi, Parthasarathy and Mishra claimed the NP-hardness of broadcasting in unit disk graphs and constructed an improved broadcasting scheme with running time at most $648D$. Very recently, the broadcasting time was further reduced to $16D - 15$ and $D + O(\log D)$ respectively by Huang et al. [18].

Gossiping in general graphs: Gossiping in wireless networks with known topology in the context of communication with arbitrarily large messages was first studied by Gąsieniec et al. in [16], where several optimal gossiping schedules were shown for a wide range of radio network topologies. For arbitrary topology wireless radio networks, an $O(D+\Delta \log n)$ schedule was given by Gąsieniec et al. in [17], where Δ is the maximum degree of the network. Very recently, Cicalese et al. [9] provided a new (efficiently computable) deterministic schedule that uses $O(D + \frac{\Delta \log n}{\log \Delta - \log\log n})$ time units to complete the gossiping task in any wireless radio network with maximum degree $\Delta = \Omega(\log n)$. Later in [25], Manne and Xin further improve the gossiping time to $O(D+\frac{\Delta \log n}{\log \Delta})$ in any wireless radio network of maximum degree $\Delta = \Omega(\log^{\frac{c}{c-1}} n)$, for any constant $c > 1$, which is an optimal schedule in the sense that there exists a network topology, specifically a Δ-regular tree, in which the gossiping cannot be completed in less than $\Omega(D + \frac{\Delta \log n}{\log \Delta})$ units of time.

M2M multicasting in general graphs: The primitive of M2M multicasting was first abstracted by Gąsieniec et al. in [13], who developed a deterministic protocol that terminates in $O(d \log^2 n + k \log^4 n)$ time. In [5], Chlebus, Kowalski and Radzik showed that the lower bound of M2M multicasting for any deterministic protocol is $\Omega(d + \frac{k \log n}{\log k})$. They also gave a randomized M2M multicasting protocol working in $O((d+k+\log^2 n) \log k)$ time. Moreover, an $O(d+k)$-time deterministic M2M multicasting protocol for the special case when the locations of the k participants are also known in advance can be found in [5] as well.

3. The preliminaries

3.1. Network model and assumptions

The WMN is modeled as an n-node undirected connected graph $G = (V, E)$ where the nodes are assigned unique labels from the set of $[n] = \{1, \ldots, n\}$. An edge $e = \langle u, v \rangle$ between u and v means that v can hear the message sent by u and vice versa. The common used unit disk model assumes that all nodes have the same transmission range and the neighbor nodes of u can hear the message sent by u as long as they are within u's transmission range. In this paper, we will first study the M2M multicasting problem in the general graphs which do not depend on such assumptions. In the general graphs to be studied, the wireless mesh nodes may have different transmission ranges and a mesh node v may not be able to hear the message from u even if v is within u's transmission range. In other words, the connectivity between two mesh nodes may depend on the physical environment and the connectivity information is given in the network topology information. We then study the M2M multicasting problem in unit disk graphs which depends on the unit disk model assumptions.

In a WMN, mesh nodes send messages in synchronous *steps* (time slots). In each step, every mesh node acts either as a *transmitter* or as a *receiver*. A mesh node acting as a transmitter sends a message which can potentially reach all of its neighbors. A mesh node acting as a receiver in a given step gets a message, if and only if, exactly one of its neighbors transmits in this step. If at least two neighbors v and v' of u transmit simultaneously in a given step, none of the messages is received by u in this step. In this case we say that a *collision* occurred at u. It is assumed that the effect at mesh node u of more than one of its neighbors transmitting is the same as that of no neighbor transmitting, i.e., the mesh nodes cannot distinguish between the collisions and the background noise.

3.2. General protocol framework

The general protocol framework consists of offline preprocessing and online M2M multicasting. The offline preprocessing is purely based on the network topology information and it does not know which mesh nodes are the participants in advance. The results of the preprocessing can be used for all M2M multicasting sessions. It includes the following tasks:

- Given G and n, the mesh nodes are organized into different clusters [13].
- Schedule the transmissions of clusters such that clusters transmitting simultaneously are at least 2 hops apart.
- In each cluster, a unique mesh node with the smallest label is elected as the leader.
- A super gathering spanning tree (SGST) is constructed for each cluster. The construction of SGST will be introduced shortly.

With the offline preprocessing, the online M2M multicasting protocol works as follows:

- Stage 1: When a cluster is scheduled to transmit, all participants of the M2M multicasting session in this cluster send their messages to the leader of the cluster according to the SGST constructed for that cluster. The leader of the cluster gets a compound message which includes all participants' initial messages in the cluster;
- Stage 2: The leader of each cluster broadcasts the compound message obtained from Stage 1 to all participants in the cluster.

With regard to the clustering algorithm [13], we note the following results about clusters of G with k participants and the maximum distance of any two participants is less than d hops.

Lemma 1. *The clusters have the following properties:*

(1) *Each cluster is a connected subgraph of G.*
(2) *The diameter of each cluster is $O(d \log n)$.*
(3) *There is an $O(\log n)$-coloring of the clusters, such that clusters having the same color are at ≥ 2 hops apart.*
(4) *There exists at least one cluster that contains all k participants and the shortest paths between them. Moreover, any other cluster containing some (or all) of the k participants, is colored differently.*

According to Lemma 1, the simultaneous execution of transmissions in different clusters having the same color does not cause any collision because all clusters of the same color are at distance at least 2-hop apart. Meanwhile, we schedule transmissions of clusters with different colors in $O(\log n)$ different phases in order to avoid collisions between clusters with different colors. Note that some participants out of k may belong to $O(\log n)$ different clusters, however there exists at least one cluster contains all k participants. If we schedule the transmission of clusters in different phases, the cluster containing all k participants will get the chance to transmit and eventually each participant can learn the information from other $k-1$ participants in at most $O(\log n)$ phases. This gives an $O(\log n)$ slowdown in comparison with an execution in a single cluster. Within each cluster of each phase, Stage 1 and Stage 2 are conducted for the message exchange among participants located at this cluster. Thus, M2M multicasting is completed. We will introduce the algorithm for Stage 1 and Stage 2 in Section 4 for general graphs and Section 5 for unit disk graphs respectively.

We now briefly introduce how to construct a SGST in a cluster at the offline preprocessing stage, which will be used in the communication strategies for Stage 1 and Stage 2 of the online M2M multicasting algorithm.

3.3. The super gathering spanning tree (SGST)

The super gathering spanning tree is a data structure that was first introduced in [9]. In the following we describe how it is constructed.

Recall the following recursive ranking procedure of mesh nodes in a tree. Start from the leaves and proceed recursively as follows. Leaves have rank 1. Next consider a mesh node v and the set Q of its children and let r_{max} and Δ be the maximum rank of the mesh nodes in Q, and maximum degree respectively. Given a fixed integer parameter $2 \leq x \leq \Delta$, if there are less than x mesh nodes in Q of rank r_{max} then set the rank of v (e.g. rank(v, x)) to r_{max}, otherwise set the rank of v to $r_{max}+1$.

For an example, see Fig. 1, where the same tree is ranked with threshold $x = 2$ and $x = 3$, respectively.

We note the following result from [9].

Lemma 2. *Let T be a tree with n mesh nodes of maximum degree Δ. Then, $r_{max}^{[x]} \leq \lceil \log_x n \rceil$, for each $2 \leq x \leq \Delta$, where $r_{max}^{[x]} = \max_{v \in T} \mathtt{rank}(v, x)$.*

For clarity of presentation, we reproduce the definitions from [9].

In each cluster constructed by our clustering approach mentioned above (e.g. see the detailed properties in Lemma 1), we can construct an arbitrary BFS spanning tree rooted at the leader λ. For the instant, we can consider an arbitrary BFS spanning tree with root λ. According to the hop distance from λ, the mesh nodes in the tree are partitioned into consecutive layers $L_i = \{v \mid dist(\lambda, v) = i\}$, for $i = 0, \ldots, r$ where r is a radius of the tree. We denote the size of each layer L_i by $|L_i|$.

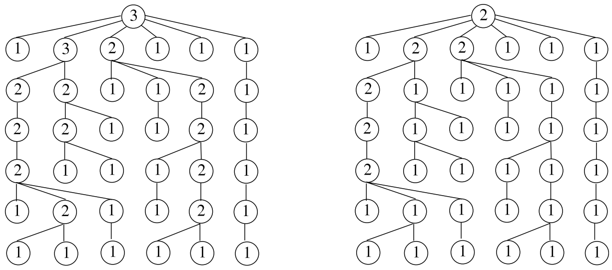

Fig. 1. A tree of size $n = 37$ ranked with $x = 2$ and $x = 3$.

For a fixed parameter $x \geq 2$, let rank set $R_j(x) = \{v \mid \mathtt{rank}(v, x) = j\}$, where $1 \leq j \leq r_{max}^{[x]}$. The optimal value of the parameter x will be determined in our M2M multicasting approaches later.

Based on the partition of the rank sets mentioned above and a fixed parameter x, the mesh nodes are divided into three different types of transmission sets.

Definition 3. The *fast transmission set* is given by $F_j^q = \{v \mid v \in L_q \cap R_j(2)$ and $parent(v) \in R_j(2)\}$. Also define $F_j = \bigcup_{q=1}^{D} F_j^q$ and $F = \bigcup_{j=1}^{r_{max}^{[2]}} F_j$.

Each mesh node in the fast transmission set has the exactly same rank as its corresponding parent in the given BFS spanning tree for the parameter $x = 2$.

Definition 4. The *slow transmission set* is given by $S_j^q = \{v \mid v \in L_q \cap R_j(2)$ and $parent(v) \in R_p(2)$, for some $p > j$; and $\mathtt{rank}(v, x) = \mathtt{rank}(parent(v), x), x > 2\}$. Also define $S_j = \bigcup_{q=1}^{D} S_j^q$ and $S = \bigcup_{j=1}^{r_{max}^{[2]}} S_j$.

Each mesh node in the slow transmission set has a smaller rank as its corresponding parent in the given BFS spanning tree for the parameter $x = 2$ however, it has a same rank as its parent in the exactly same BFS spanning tree for a given fixed parameter $x > 2$. Note also that the optimal value of x will be determined later in our M2M multicasting algorithms.

Definition 5. The *super-slow transmission set* is given by $SS_j^q = \{v \mid v \in L_q \cap R_j(x)$ and $parent(v) \in R_{j'}(x), j' > j\}$. Accordingly, define $SS_j = \bigcup_{q=1}^{D} SS_j^q$ and $SS = \bigcup_{j=1}^{r_{max}^{[x]}} SS_j$.

Similarly, each mesh node in the super-slow transmission set has a smaller rank as its corresponding parent in the BFS spanning tree for a given fixed parameter $x > 2$.

Note that the above transmission sets define a partition of the mesh node set. An example of the partition can be found in Fig. 2. Each mesh node v only belongs to one of the transmission sets and $V = F \bigcup S \bigcup SS$.

A ***super-gathering spanning tree*** (*SGST*) for a graph $G = (V, E)$ is any BFS spanning tree T_G of G, ranked according to the ranking procedure above and satisfying: (1) T_G is rooted at an arbitrary pre-determined mesh node λ of G; (2) T_G is ranked; (3) all mesh nodes in F_j^q of T_G are able to transmit their messages to their parents simultaneously without any collision, for all $1 \leq q \leq D$ and $1 \leq j \leq r_{max}^{[2]} \leq \lceil \log n \rceil$; (4) every mesh node v in $S_j^q \cap R_{j'}(x)$ of T_G has following property: $parent(v)$ has at most $x - 1$ neighbors in $S_j^q \cap R_{j'}(x)$, for all $j' = 1, 2, \ldots, r_{max}^{[x]} \leq \lceil \log_x n \rceil, j = 1, 2, \ldots, r_{max}^{[2]} \leq \lceil \log n \rceil$ and $q = 1, \ldots, D$.

An exact tree topology (e.g., subclass of the graphs) gives more intuition of our new data structure *SGST* in which the parent of a mesh node in the fast transmission set has the exact one child in the corresponding fast transmission set whereas it has a bounded number of children (e.g., upper bounded by x) in the slow transmission set and has more than x children for the mesh nodes in the super-slow transmission set. It is worth to remarking that it has been shown that a *SGST* can be always constructed from the given graph in polynomial time in terms of the network size.

An example of *SGST* constructed from the original graph is given in Fig. 2.

The existence of a *SGST* for any graph G was shown in [9] by the following theorem.

Theorem 6. *For an arbitrary graph G, there exists an $O(n^2 \log n)$ time construction of a SGST.*

3.4. The time complexity of the preprocessing stage

The preprocessing stage is used to group mesh nodes into clusters, generate scheduling among clusters, and construct SGST for each cluster with various parameters d and k in a given WMN of size n in advance without any knowledge on the locations of the participants, where d is the maximum hop distance between any two out of the k participating mesh nodes. Such a preprocessing stage will be performed offline only once. After that, the execution of the online M2M multicasting schedule with given parameters d and k can be done immediately without any extra time to construct the communication strategy.

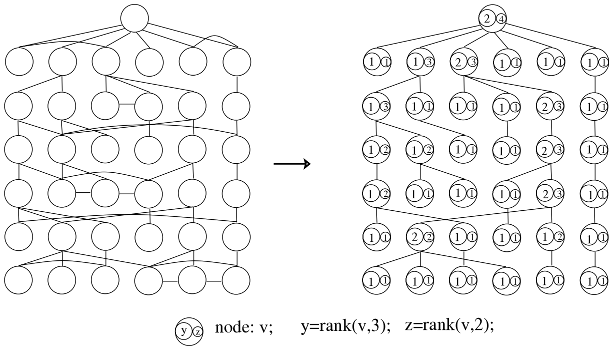

Fig. 2. From the original graph to a super-gathering-spanning-tree .

Theorem 7. *The preprocessing stage for the given parameters d and k in a given WMN of size n can be constructed in time $O(n^3)$ in advance without knowing the locations of the k participating mesh nodes.*

Proof. According to the construction of the clusters in [13] and Lemma 13 in [14], we can know that the clusters we constructed can be computed in time $O(n^3)$, which includes the time for construction of all clusters and the time for coloring the clusters. Combining with the result from Theorem 6, it completes the proof. □

Consequently, for a given WMN of size n, we can do the preprocessing in advance for all combinations of various parameters d, k in time $O(Dn^4)$ by a brute force fashion, where D is the diameter of the WMN, $2 \leq k \leq n$ is the number of participating mesh nodes, $1 \leq d \leq D$ is the maximum hop distance between any two out of the k participants.

Theorem 9. *The offline preprocessing stage for all combinations of parameters d, k in a given WMN of size n can be constructed in time $O(Dn^4)$.*

Proof. The number of different combinations with the parameters d, k can be bounded by Dn since $1 \leq d \leq D$ and $2 \leq k \leq n$. Therefore, combining with Theorem 7, the theorem directly follows. □

4. Deterministic M2M multicast in general graphs

Without loss of generality, we assume the initial message held by each participant is the label of the mesh node. The aim of the algorithm is that each participant learns the labels of all other participants.

As stated in the general protocol framework, our M2M multicasting protocols within a single cluster needs to conduct two stages, Stage 1 which is to gather the labels of all participants located at the same cluster to the leader of the cluster, called *CONVERGECAST-SGST* stage; and Stage 2 which is to broadcast the compound message at the leader to all mesh nodes in the cluster, called *BROADCAST* stage.

We use the notation $T_C(n, d, k)$ and $T_B(n, d, k)$ to denote the number of rounds used by CONVERGECAST-SGST stage and BROADCAST stage in a single cluster, respectively. It is then clear that our M2M protocol solves the online M2M multicasting problem in time $O((T_C(n, d, k) + T_B(n, d, k)) \log n)$. Thus, the following theorem holds.

Theorem 11. *If CONVERGECAST-SGST and BROADCAST protocols are used in M2M multicasting for a single cluster to complete the message exchange among the participants within the same cluster, then M2M multicasting completes message exchange of k participants in time $O((T_C(n, d, k) + T_B(n, d, k)) \log n)$.*

4.1. The CONVERGECAST-SGST stage

In this section, we show how the labels of the participants can be gathered at the leader λ (one unique mesh node with the smallest label) in each cluster C efficiently based on a *super gathering spanning tree* (SGST) computed in the offline preprocessing stage. Note that the diameter of the cluster C is upper bounded by $O(d \log n)$, where d and n are the diameter among all k participants and the size of the network respectively.

The communication process will be split into consecutive blocks of 9 time units each. The first 3 units of each block are used for fast transmissions from the set F, the middle 3 units are reserved for slow transmissions from the set S and the remaining 3 are used for super-slow transmissions of mesh nodes from the set SS. We use 3 units of time for each type of transmission in order to prevent collisions between neighboring BFS layers.

According to property (3) of the SGST, all mesh nodes in F_j^q are able to transmit their messages to their parents simultaneously without any collision, for all $1 \leq q \leq D$ and $1 \leq j \leq r_{max}^{[2]} \leq \lceil \log n \rceil$, which will be used in our communication strategy later.

Recall that we can move all messages stored in $S_j^q \cap R_{j'}(x)$ to their parents in $SGST$ within time $x - 1$ due to Lemma 4 in [16] together with property(4) of the SGST, where x is a constant integer, $1 \leq j \leq r_{max}^{[2]}$, $1 \leq j' \leq r_{max}^{[x]}$, and $1 \leq q \leq D$. The optimal value of x will be determined later.

For all $j = 1, 2, \ldots, r_{max}^{[2]} \le \lceil \log n \rceil$, $j' = 1, 2, \ldots, r_{max}^{[x]} \le \lceil \log_x n \rceil$ and $q = 1, \ldots, D$, we can compute for each node $v \in S_j^q \cap R_{j'}(x)$ at layer q the number of a step $1 \le s(v) \le x - 1$ in which mesh node v can transmit without interruption from other mesh nodes in $S_j^q \cap R_{j'}(x)$ which are also at layer q. Due to our technical purpose, we set $x = k$. This also means that $1 \le s(v) \le k - 1$.

In what follows, we show how to deliver the messages from the mesh nodes in super-slow transmission sets SS to their parents in *SGST* efficiently, which depends on how we resolve collision. As shown in [8,7], the most efficient tools designed for collision resolution are based on combinatorial structures possessing a *selectivity property*. We say that a set R hits a set Z on element z, if $R \cap Z = \{z\}$, and a family of sets $\mathcal{F}$ hits a set Z on element z, if $R \cap Z = \{z\}$ for at least one $R \in \mathcal{F}$. In [8] a family of subsets of the set $\{1, 2, \ldots, n\} \equiv [n]$ is defined that hits each subset of $[n]$ of size at most $k \le n$ on all of its k elements. This family of subsets is referred to as being *strongly k-selective*. It is also shown that there exists such a family of size $O(k^2 \log n)$. The work presented in [7] defines a family of subsets of the set $\{1, 2, \ldots, n\} \equiv [n]$ which hits each subset of $[n]$ of size at most k on at least $k/2$ distinct elements, where $1 \le k \le n$. This family is referred to as a *k-selector* and such a family of size $O(k \log n)$ is shown to exist.

In the following we show how to cope with collisions that occur during the competition process through the use of selective families and selectors.

Assume that we have a connected bipartite graph B in which mesh nodes are partitioned into two sets U and L. In our further considerations, the sets U and L will correspond to two adjacent BFS levels of *SGST*, upper and lower respectively, in a subgraph of G. While, in general, mesh nodes in U and L are not aware of the presence of each other, we assume here that each mesh node $v \in L$ is associated with exactly one of its neighbors $u \in U$ (later labeled as the *parent* of v) and that this relation is known to both of them. Note that a mesh node in U can be the parent of several mesh nodes in L. We assume also, that initially only mesh nodes in L are aware of their parents presence in B, i.e., their parents must be informed about this relationship by their children since the participating mesh node does not know the labels of other participants. In what follows we show how to move k messages that are available at mesh nodes of L, to the parent mesh nodes in U in time $O(k \log n)$.

It is known, that a communication mechanism based on the selector idea allows a fraction (e.g., a half) of the mesh nodes in L to deliver their messages to their parents in U in time $O(k \log n)$ [7]. Let $S(k)$ represent the collision resolution mechanism based on selectors. Note that $S(k)$, if applied in undirected networks, can be supported by an *acknowledgment of delivery* mechanism in which each transmission from the participating mesh nodes in L is alternated with an acknowledgment message coming from the parent mesh node $u \in U$. If during the execution of $S(k)$ a transmission from v towards u is successful, i.e., one of u neighbors succeeds in delivering its message, the acknowledgment issued by u and returned to v confirms the successful transmission; otherwise the acknowledgment is null. Let $\mathbf{S}(k)$ be the mechanism with this acknowledgment feature added to $S(k)$. In other words, the use of $\mathbf{S}(k)$ allows us to exclude all mesh nodes in L that have managed to deliver their message to their parent in U during the execution of $\mathbf{S}(k)$ from further transmissions. Note that the duration of $\mathbf{S}(k)$ is $O(k \log n)$, see [7].

Let $S^*(i)$ be the communication mechanism based on the concatenation (superposition) of i selectors $S(2^i), S(2^{i-1}), \ldots, S(2^1)$. We call this a *descending selector*. The descending selector extended by the acknowledgment mechanism, i.e., the concatenation of $\mathbf{S}(2^i), \mathbf{S}(2^{i-1}), \ldots, \mathbf{S}(2^1)$, forms *a promoter* and it is denoted by $\mathbf{S}^*(k)$. Note that the duration of $\mathbf{S}^*(k)$ is $O(k \log n)$.

Lemma 12. *All messages from the k participants can be collected from one partition of a bipartite graph to another partition in time $O(k \log n)$.*

Proof. The proof is done by induction, and is based on the fact that after the execution of each $\mathbf{S}(2^j)$, for $j = \lceil \log k \rceil, \ldots, 1$, the number of competing nodes in L is bounded by 2^{j-1}. $\square$

According to Lemma 12, we could also compute for each mesh node $u \in SS_i$ at layer q the step number $1 \le ss(u) \le ck \log n$ for some constant integer $c \ge 1$ in which the mesh node u can transmit without interruption from other mesh nodes in SS_i also in layer q.

Let v be a node at layer q and with $\mathtt{rank}(v, 2) = j$ and $\mathtt{rank}(v, k) = i$, in SGST of a cluster C. Further, let $d' = O(d \log n)$ be the diameter of the cluster C.

To simplify our presentation, let $\Re(d', q, k, n) = (d' - q + 1) + (j - 1)(k - 1) + (i - 1)(c + 1)k\lceil \log n \rceil$. Depending on if v belongs to the set F, to the set S or to the set SS, it will transmit in the time block $t(v)$ given by:

$$t(v) = \begin{cases} \Re(d', q, k, n) & \text{if } v \in F \\ \Re(d', q, k, n) + s(v) & \text{if } v \in S \\ \Re(d', q, k, n) + ss(v) & \text{if } v \in SS. \end{cases} \tag{1}$$

The details of our algorithm for the CONVERGECAST-SGST stage are illustrated in **Algorithm 1**.

We observe that any mesh node v in the *SGST* requires at most d' fast transmissions, $\log n$ slow transmissions and $\log_k n$ super-slow transmissions to deliver its message to the root (the leader λ) of the *SGST* if there is no collision during each transmission. Moreover, the above definition of $t(v)$ results in the following lemma.

Lemma 14. *A mesh node v transmits its message as well as all messages collected from its descendants towards its parent in SGST successfully during the time block allocated to it by the transmission pattern.*

Algorithm 1 The approach for CONVERGECAST-SGST stage in a single cluster for general graphs

Input: the cluster C, the number of participants k, diameter among the participants d, a SGST rooted at λ based on the fixed parameter $x = k$.

for $q = 1$ to D **do**
  for $j = 1$ to $r_{max}^{[2]} \leq \lceil \log n \rceil$ **do**
    for each node $v \in S_j^q$ **do**
      compute the number of a step $1 \leq s(v) \leq k - 1$ according to Lemma 4 in [16] together with property(4) of the SGST;
for $q = 1$ to D **do**
  for $i = 1$ to $r_{max}^{[k]}$ **do**
    for each node $u \in SS_i^q$ **do**
      compute the number of a step $1 \leq ss(u) \leq ck \log n$ for some constant integer $c \geq 1$ according to Lemma 12;
Let $d' = O(d \log n)$ be the diameter of the cluster C;
$timeBlockCounter = 0$;// counter for time blocks used for transmissions
repeat
  for $counter_1 = 1$ to 3 **do**
    // avoid the transmissions from different types of transmission sets
    for $counter_2 = 0$ to 2 **do**
      // avoid the transmissions from the consecutive BFS layers
      $timeBlockCounter = timeBlockCounter + 1$;
      for each $v \in L_q$, where $1 \leq q \leq D$ **do**
        $j = rank(v, 2)$;
        $i = rank(v, k)$;
        $\Re(d', q, k, n) = (d' - q + 1) + (j - 1)(k - 1) + (i - 1)(c + 1)k\lceil \log n \rceil$;
        if $((v \in F) \wedge (counter_1 = 1)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter)$ **then**
          v transmits;
        if $((v \in S) \wedge (counter_1 = 2)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter + s(v))$ **then**
          v transmits;
        if $((v \in SS) \wedge (counter_1 = 3)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter + ss(v))$ **then**
          v transmits;
until λ gets all messages from the participants in the cluster C;

Proof. Let v be a mesh node at layer q such that $\texttt{rank}(v, 2) = j$ and $\texttt{rank}(v, k) = i$. For each mesh node w at layer $q' > q$, which is a descendant of v we have that $\texttt{rank}(w, 2) = j' \leq j = \texttt{rank}(v, 2)$ and $\texttt{rank}(w, k) = i' \leq i = \texttt{rank}(v, k)$. Therefore if $v, w \in F$, the first term of the expression $(d' - q' + 1) + (j' - 1)(k - 1) + (i' - 1)(c + 1)k\lceil \log n \rceil$ is smaller for w. Hence, according to the pattern of transmissions above, it is not hard to see that mesh node w transmits earlier than mesh node v also holds for other cases (e.g. $v \in SS$ and $w \in F$).

We now prove that any mesh node v following the pattern of transmissions will transmit to its parent without being interrupted by anyone else.

In fact, no collision can happen between neighboring BFS layers because of the separation into three subsequences, ensuring that three time units are available within each block. Nor can there be collisions between transmissions coming from different transmission sets (fast, slow and super-slow), because of the three parts of each time block. It remains to rule out collisions between mesh nodes within the same transmission sets and at the same BFS layer in the SGST.

Assume that $v, w \in F$ and that they are at the same BFS layer q in the SGST.

If v and w also have the same rank $\texttt{rank}(v, 2) = \texttt{rank}(w, 2)$ and $\texttt{rank}(v, k) = \texttt{rank}(w, k)$, then they do not interrupt each other due to the properties of SGST.

If they have different ranks $\texttt{rank}(v, 2) = j \neq j' = \texttt{rank}(w, 2)$ but $\texttt{rank}(v, k) = \texttt{rank}(w, k)$ or $\texttt{rank}(v, k) = i \neq i' = \texttt{rank}(w, k)$ but $\texttt{rank}(v, 2) = \texttt{rank}(w, 2)$ respectively, then they transmit in different time blocks according to the pattern of transmissions for the mesh nodes in F.

If $\texttt{rank}(v, 2) = j \neq j' = \texttt{rank}(w, 2)$ and $\texttt{rank}(v, k) = i \neq i' = \texttt{rank}(w, k)$, the transmission pattern separates v, w by at least $|(j - j') \cdot (k - 1) + (i - i')(c + 1)k\lceil \log n \rceil| > (c + 1)k\lceil \log n \rceil - \log n(k - 1) > ck\lceil \log n \rceil$ time blocks. The inequalities follow since $|j - j'| \leq \log n$, and $|i - i'| \leq 1$. Consequently, v and w cannot interfere with each other, either.

Assume now that $v, w \in S$ and that they are at the same BFS layer q in GST.

If $\texttt{rank}(v, k) = \texttt{rank}(w, k)$, then either $\texttt{rank}(v, 2) = j \neq j' = \texttt{rank}(w, 2)$ and both $s(v), s(w) \leq k$, or if they have the same rank j, then they have different values of $s(v)$ and $s(w)$. Hence, they do not interrupt each other.

If $\texttt{rank}(v, 2) = j \neq j' = \texttt{rank}(w, 2)$ and $\texttt{rank}(v, k) = i \neq i' = \texttt{rank}(w, k)$, the pattern of transmissions separates v, w by at least $|(j - j') \cdot (k - 1) + (s(v) - s(w)) + (i - i')(c + 1)k\lceil \log n \rceil| > (c + 1)k\lceil \log n \rceil - \log n(k - 1) - (k - 2) > 1$ time blocks. The inequalities follow since $|j - j'| \leq \log n$, $|i - i'| \geq 1$, and $|s(v) - s(w)| \leq k - 2$. Thus, there cannot be a collision between v and w.

Using similar arguments, we can also prove that when $v, w \in SS$ no collision can happen either. This completes the proof. □

Since the number of time blocks used is no more than $d' + (k-1)\lceil \log n \rceil + (c+1)k\lceil \frac{\log^2 n}{\log k} \rceil$, we have

Lemma 16. $T_C(n, d, k) = O(d' + \frac{k \log^2 n}{\log k}) = O(d \log n + \frac{k \log^2 n}{\log k})$.

4.2. The BROADCAST stage

The distribution of the compound message is broadcast by the leader λ of the cluster to all participants in the same cluster C by reversing the direction of the transmissions in the CONVERGECAST-SGST stage. This can be achieved in time $O(d \log n + \frac{k \log^2 n}{\log k})$. Therefore, the following lemma directly follows.

Lemma 17. $T_B(n, d, k) = O(d \log n + \frac{k \log^2 n}{\log k})$.

Combining the results Theorem 6, Theorem 11, Lemma 16, and Lemma 17, we get the desired result.

Theorem 18. *The M2M multicasting problem in arbitrary WMNs can be solved in time* $O(d \log^2 n + \frac{k \log^3 n}{\log k})$.

5. Deterministic M2M multicast in unit disk graphs

In this section, we first introduce the model we employed for the WMN. Then, we give a time efficient M2M multicasting scheme with running time $O(d \log^2 n + k \log n)$ based on the general protocol framework in Section 3.2. Finally, we show a new distributed algorithm which accomplishes the online M2M multicasting in time $O(d+k)$. This is asymptotically optimal in the sense that there exists a WMN topology, e.g. a line, a ring, a star or a complete graph, in which the M2M multicasting cannot be completed in less than $\Omega(d+k)$ units of time.

5.1. The model

We consider a WMN which is modeled as an undirected connected graph $G = (V, E)$, where V represents the set of mesh nodes in the WMN which arbitrarily distributed in the Euclidean plane R^2, and E contains unordered pairs of distinct mesh nodes, such that $(v, w) \in E$ iff the transmissions of mesh node v can directly reach mesh node w and vice versa (the reachability of transmissions is assumed to be a symmetric relation). In this case, we say that the mesh nodes v and w are *neighbors* in G. We assume that every mesh node in the WMN has the same transmission range.

Due to the interference constraint, the distance between any two communicating mesh nodes shall be greater than a constant minimum bound, say ϵ, which was called $\Omega(1)$ model in [20]. In addition, the distance shall be smaller than a constant in order to maintain the connectivity of G. Without loss of generality, the constant can be set as a unit. As a consequence, the Unit Disk Graph model can be adopted for the WMN [6,21]. In the following, the terms between unit disk graph and the WMN may be used interchangeable.

The *degree* of a mesh node w is its number of neighbors. We use Δ to denote the *maximum degree* of the WMN, i.e., the maximum degree of any mesh node in the WMN. The *size of the network* is the number of mesh nodes $n = |V|$. Communication in the WMN is synchronous and consists of a sequence of communication steps, which is the same assumption we used in the general WMNs.

5.2. $O(d \log^2 n + k \log n)$-time M2M multicasting

Our new M2M multicasting protocol based on same approaches (clustering methods and the SGST) as described in Section 4. We observe some good properties in the unit disk graphs which can be used to improve the time complexity of $T_C(n, d, k)$ in the CONVERGECAST-SGST stage. This enables us to achieve a better M2M multicasting schedule in time $O(d \log^2 n + k \log n)$.

Lemma 19. *In a unit disk graph, all mesh nodes in set S_i of the SGST can transmit their messages to the corresponding parents in time $O(1)$.*

Proof. In [16], it states that all messages from the mesh nodes in one partition can be moved to another partition in a bipartite graph (in this case two consecutive BFS layers) in Δ time units, where Δ is the maximum degree of the graph. The solution is based on a notion of the *minimal covering sets*. Furthermore in [20], it proves that unit disk graphs we employed here have the property of bounded degree $O(1)$. The lemma follows. □

Using the same arguments, we can also show the following lemma.

Lemma 21. *In a unit disk graph, all mesh nodes in set SS_i of the SGST can transmit their messages to the corresponding parents in time $O(1)$.*

Similarly, we can compute the numbers of the steps $s(v)$ and $ss(u)$ for $v \in S$ and $u \in SS$.

Moreover, we can also define a transmission pattern $t(v)$ in terms of the time block for any mesh node v in the given unit disk graph.

$$t(v) = \begin{cases} \Re(d', q, k, n) & \text{if } v \in F \\ \Re(d', q, k, n) + s(v) & \text{if } v \in S \\ \Re(d', q, k, n) + ss(v) & \text{if } v \in SS \end{cases} \tag{2}$$

where $\Re(d', q, k, n) = (d' - q + 1) + (j - 1)A + (i - 1)B$, $A \ll B = O(1)$ are the constant integers that are determined according to Lemma 19 and Lemma 21.

The details of the approach for the CONVERGECAST-SGST stage are illustrated in **Algorithm 2**. Note that the main difference with Algorithm 1 is that both the numbers of a step $s(v)$ and a step $ss(u)$ can be bounded by a constant integer due to Lemma 19 and Lemma 21, where $v \in S$ and $u \in SS$.

Algorithm 2 The approach for CONVERGECAST-SGST stage in a single cluster for unit disk graphs

Input: the cluster C, the number of participants k, diameter among the participants d, a SGST rooted at λ based on the fixed parameter $x = k$.

for $q = 1$ to D **do**
  for $j = 1$ to $r_{max}^{[2]} \le \lceil \log n \rceil$ **do**
    for each node $v \in S_j^q$ **do**
      compute the number of a step $1 \le s(v) \le A = O(1)$ for a constant integer A according to Lemma 19;
for $q = 1$ to D **do**
  for $i = 1$ to $r_{max}^{[k]}$ **do**
    for each node $u \in SS_i^q$ **do**
      compute the number of a step $1 \le ss(u) \le B = O(1)$ for a constant integer B according to Lemma 21;
Let $d' = O(d \log n)$ be the diameter of the cluster C;
timeBlockCounter $= 0$;// counter for time blocks used for transmissions
repeat
  for $counter_1 = 1$ to 3 **do**
    // avoid the transmissions from different types of transmission sets
    for $counter_2 = 0$ to 2 **do**
      // avoid the transmissions from the consecutive BFS layers
      $timeBlockCounter = timeBlockCounter + 1$;
      for each $v \in L_q$, where $1 \le q \le D$ **do**
        $j = rank(v, 2)$;
        $i = rank(v, k)$;
        $\Re(d', q, k, n) = (d' - q + 1) + (j - 1)A + (i - 1)B$;
        if $((v \in F) \wedge (counter_1 = 1)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter)$ **then**
          v transmits;
        if $((v \in S) \wedge (counter_1 = 2)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter + s(v))$ **then**
          v transmits;
        if $((v \in SS) \wedge (counter_1 = 3)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter + ss(v))$ **then**
          v transmits;
until λ gets all messages from the participants in the cluster C;

In the CONVERGECAST-SGST stage, the labels of all the participants will be gathered at the leader λ of the cluster C based on a SGST computed in advance. Thanks to the time division scheme (see Section 4.1), it guarantees there is no any collision from different types of transmission (F, S and SS). And also no collision can occur between the transmissions from the consecutive BFS layers. Lemma 19 (Lemma 21) shows that the collision by the competing mesh nodes from the slow transmission set S (the super slow transmission set SS) can be solved in time $O(1)$. Furthermore, we observe that it needs at most d' fast transmissions, k slow transmissions and k super-slow transmissions to forward the message from any mesh node to λ on the SGST in the cluster C, where $d' = O(d \log n)$ is the diameter of C. Consequently, $T_C(n, d, k) = O(d' + k) = O(d \log n + k)$. Moreover, the distribution of the compound message can be broadcasted by the mesh node λ to the other mesh nodes in the cluster C by reversing the direction of the transmissions in the CONVERGECAST-SGST stage. This implies that $T_B(n, d, k) = O(d \log n + k)$. Combining the results Lemma 1, Theorem 11, Theorem 6, Lemma 19, and Lemma 21, we get the desired result.

Theorem 22. *The M2M multicasting problem in the unit disk graph can be solved in time $O(d \log^2 n + k \log n)$.*

5.3. $O(d + k)$-time M2M multicast

In the previous sections, the transmission process was split into separate $O(\log n)$ phases according to coloring of the clusters, each costing $(T_C(n, d, k) + T_B(n, d, k))$ units of time. In this section we show how to pipeline the transmissions of different phases. This will allow a new online M2M multicast schedule of length $O(d+k)$. The new schedule based on a new cluster method and a well-known scheme of the vertex coloring in the unit disk graphs.

5.3.1. Graph clustering preserving locality

As mentioned before, the main purpose of the clustering method is to obtain a representation of a large graph as a collection of its much smaller subgraphs (clusters), while preserving local distances between the mesh nodes.

Our new clustering method groups the mesh nodes belonging to some connected subgraphs G' into the same cluster C. If the diameter of G' is d, the diameter of C is at most $O(d)$ which improves the stretch of the clusters [13] by $O(\log n)$ factor. (See Lemma 1 for the details.)

Given a graph G, we partition the mesh nodes to different BFS levels starting from an arbitrary mesh node λ. To simplify our presentation, we use the same definition of the graph partition as in [13].

Definition 23. A *partition* $\pi(x)$ of the graph G is a division of G into super-levels, such that, each super-level is composed of $4d$ consecutive BFS levels, where the first super-level starts from an arbitrary but fixed BFS level L_x (note that levels $L_0, L_1, \dots, L_{x-1}$ are excluded from the partition $\pi(x)$). More formally, the ith super-level in $\pi(x)$ is $G_i(x) = \{v | v \in L_j, (i-1-x)\cdot 4d \leq j \leq (i-x)\cdot 4d - 1\}$, for $i = 1, 2, \cdots, \lceil \frac{D-x}{4d} \rceil$, where D is the radius of G with respect to arbitrary node λ. Given a super-level $G_i(x)$, its *top* level is $L_{(i-1-x)\cdot 4d}$, and its *bottom* level is $L_{(i-x)\cdot 4d-1}$. Note that $G_i(x)$ is not necessarily connected.

Definition 24. For each mesh node u belonging to the top level of $G_i(x)$, we define the *cluster* $C_u^{(i)}$, which contains all mesh nodes in $G_i(x)$ at distance $\leq 4d$ from u.

Lemma 25. *The clusters have the following property: the diameter of each cluster is bounded by $O(d)$.*

Proof. Property follows directly from the construction of the clusters. □

Note that the construction of the cluster we used here is different and much simpler than the previous work in [13]. Moreover, the stretch of the clusters in term of the diameter is better.

Definition 27. The 2-*partition* of the graph G comprises two different partitions: $\pi(0)$ which starts at the super-level $G_1(0)$, and $\pi(2d)$ which starts at the super-level $G_1(2d)$.

Using the same arguments from [13] but for different construction of the clusters, we can show the following lemma.

Lemma 28. *In at least one of the partitions of the* 2-*partition, there exists at least one cluster that contains all k participating mesh nodes and the shortest paths between them.*

Proof. Let v be one of the k participants. According to our definition of the 2-*partition*, we can prove that the mesh node v must fall into the central $2d$ BFS levels of a super-level in one of the partitions, except for the case when v belongs to the first d BFS levels (when all k participants belong to the cluster based on the pre-selected mesh node λ). Thus, there exists a mesh node p at the top level of the corresponding super-level $G_i(\cdot)$, which is at distance $dist(p, v) \leq 3d$ from the mesh node v. Since all other participating mesh nodes are at distance $\leq d$ from v, there exists a cluster $C_p^{(i)}$ which contains the entire set of k participating mesh nodes. □

5.3.2. M2M multicast in a single cluster

By employing the same strategy of M2M multicast in a single cluster in Section 5.2 and property of the clusters (Lemma 25), the following lemma directly follows.

Lemma 30. *In a single cluster, the M2M multicasting problem in the unit disk graph can be solved in time $O(d+k)$.*

The details of M2M multicasting scheme in a single cluster are illustrated in **Algorithm 3**. Note that the main difference with Algorithm 2 is that the new clusters used for the CONVERGECAST-SGST stage have asymptotical optimal diameter which is $O(d)$, where d is the original diameter among the all k participants.

5.3.3. Distance-2 vertex coloring

In distance-2 vertex coloring scheme, vertices separated by a distance of less than or equal to two hops must receive different colors. This scheme will be used for our new M2M multicast schedule later. The following lemma had been stated in [22].

Lemma 31. *Distance-2 vertex coloring in general graphs can be solved in $O(\Delta^2)$ colors, where Δ is the maximum degree of the graph.*

Due to the "special property" of the unit disk graphs, the following lemma can be derived.

Lemma 32. *Distance-2 vertex coloring in unit disk graphs can be solved in $O(1)$ colors.*

Proof. It had been show that unit disk graph model employed in our work has the property of bounded degree $O(1)$ in [20]. Combining with Lemma 31, it completes the proof. □

Algorithm 3 Faster M2M multicasting scheme in a single cluster for unit disk graphs based on a new clustering approach in Section 5.3.1

Input: the cluster C, the number of participants k, diameter among the participants d, a SGST rooted at λ based on the fixed parameter $x = k$.

Step 1: CONVERGECAST-SGST stage
for $q = 1$ to D **do**
  for $j = 1$ to $r_{max}^{[2]} \leq \lceil \log n \rceil$ **do**
    for each node $v \in S_j^q$ **do**
      compute the number of a step $1 \leq s(v) \leq A = O(1)$ for a constant integer A according to Lemma 19;
for $q = 1$ to D **do**
  for $i = 1$ to $r_{max}^{[k]}$ **do**
    for each node $u \in SS_i^q$ **do**
      compute the number of a step $1 \leq ss(u) \leq B = O(1)$ for a constant integer B according to Lemma 21;
Let $d' = O(d)$ **be the diameter of the cluster** C;
$timeBlockCounter = 0$;// counter for time blocks used for transmissions
repeat
  for $counter_1 = 1$ to 3 **do**
    // avoid the transmissions from different types of transmission sets
    for $counter_2 = 0$ to 2 **do**
      // avoid the transmissions from the consecutive BFS layers
      $timeBlockCounter = timeBlockCounter + 1$;
      for each $v \in L_q$, where $1 \leq q \leq D$ **do**
        $j = rank(v, 2)$;
        $i = rank(v, k)$;
        $\Re(d', q, k, n) = (d' - q + 1) + (j - 1)A + (i - 1)B$;
        if $((v \in F) \wedge (counter_1 = 1)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter)$ **then**
          v transmits;
        if $((v \in S) \wedge (counter_1 = 2)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter + s(v))$ **then**
          v transmits;
        if $((v \in SS) \wedge (counter_1 = 3)) \wedge (q \bmod 3 \equiv counter_2) \wedge (\Re(d', q, k, n) = timeBlockCounter + ss(v))$ **then**
          v transmits;
until λ gets all messages from the participants in the cluster C;
Step 2: BROADCAST stage
reverse the direction of the transmissions in the CONVERGECAST-SGST stage;

5.3.4. M2M multicast schedule

In our new multicast schedule, we built the distance-2 coloring scheme of the mesh nodes (Lemma 32) into the time division approaches we developed for M2M multicasting problem in a single cluster in Section 5.3.2. Let $Max_c = O(1)$ denote the number of colors which used to solve distance-2 vertex coloring problem in the unit disk graph. Now we extend each time block to a time region that contains Max_c different time blocks. The ith time block is used for the transmissions from the mesh nodes with color i, where $1 \leq i \leq Max_c$. Consequently, when a mesh node v transmits at a given step, the message will be received by all neighbors of v successfully due to the property of distance-2 vertex coloring scheme, which allows v to forward the messages to different clusters simultaneously without any collisions although v may belong to at most $O(n)$ different clusters. The modified time scheme implies a new M2M multicasting schedule with a $O(Max_c)$ slowdown in comparison with an execution of M2M multicasting scheme in a single cluster. Combining with Lemma 30, we derive our main result for the unit disk graphs.

Theorem 34. *The M2M multicasting problem in the unit disk graphs can be solved in time $O(d + k)$.*

6. M2M multicasting with an unknown *d*

In this section, we show how our M2M multicasting algorithms proposed for the scenarios with knowing the diameter among the participants can be extended to the cases when the maximum hop distance d between the participants is not known in advance with asymptotically similar running time by using the standard doubling methodology.

Let M2M-Multicast-GG(d, k, n) denote the algorithm we proposed for M2M multicasting in general graphs with the known diameter d among the k participants. Our new M2M multicasting scheme for the scenario with a unknown diameter d can be described as follows (see details in **Algorithm 4**).

When the diameter d among all k participants is known in advance, our M2M multicasting scheme M2M-Multicast-GG(d, k, n) can accomplish the task in time $O(d \log^2 n + \frac{k \log^3 n}{\log k})$ due to Theorem 35. From the construction of the clusters to contain all k participants, we know that there always exists a cluster that can contain all participants when we use a

Algorithm 4 M2M multicasting scheme for general graphs with a unknown diameter among the participants

$TryDiameter = 1$; // assume that $d = 1$
repeat
    M2M-Multicast-GG($TryDiameter$, k, n);
    $TryDiameter = 2 * TryDiameter$;
until M2M multicasting task is accomplished for k participants

larger diameter than the real one among the k participants. Combining with the standard doubling approach, the total communication steps can be bounded by $O(\sum_{0\le i\le\lceil\log 2d\rceil} 2^i \log^2 n + \lceil\log 2d\rceil \frac{k\log^3 n}{\log k})$ which is $O(d\log^2 n + \log d\frac{k\log^3 n}{\log k})$. Consequently, we derive the following Theorem.

Theorem 35. *The M2M multicasting problem in arbitrary WMNs with a unknown diameter d among k participants can be solved in time* $O(d\log^2 n + \log d\frac{k\log^3 n}{\log k})$.

Similarly, we can also extend our M2M multicasting scheme to unit disk graphs with a unknown diameter among the participants.

7. Conclusion

In this paper we have shown an $O(d\log^2 n + \frac{k\log^3 n}{\log k})$-time algorithm for solving the M2M multicasting problem for a group of k participating mesh nodes each within distance d of each other, in an arbitrary WMN consisting of n mesh nodes, which is an almost optimal scheme due to the lower bound $\Omega(d + \frac{k\log n}{\log k})$ by Chlebus, Kowalski, and Radzik in [5]. This also improves the currently best known result with running time $O(d\log^2 n + k\log^4 n)$, by Gąsieniec, Kranakis, Pelc, and Xin in [13]. In this paper, we also show the M2M multicasting problem can be solved in time $O(d + k)$ in the unit disk graphs, which is asymptotically optimal. Interesting problems left for further investigation include (1) improving the upper bounds of our algorithm, (2) developing locality-sensitive multicasting algorithms for the case when the mesh nodes of the WMN have only limited (e.g., local) knowledge of the topology, (3) investigating how efficient updating affects performance of multicasting in mobile WMNs, and (4) evaluating the performance of our algorithm in real WMNs.

It is worth to remarking that our M2M multicasting algorithms can be extended to the scenario when the maximum hop distance d between the participants is not known in advance with asymptotically similar running time by using the standard doubling methodology. Note also that it would not be difficult to increase the level of fault-tolerance in our algorithm at the expense of some extra time consumption by using a tuned version of the fault-tolerant scheme proposed in [26].

Acknowledgment

Part of first author's research was performed while this author was a senior research fellow at Université catholique de Louvain, Belgium, a scientist/postdoctoral research fellow at Simula Research Laboratory and a postdoctoral research fellow at The University of Bergen, Norway. This work was partially supported by Visiting Scholarship of State Key Laboratory of Power Transmission Equipment & System Security and New Technology (Chongqing University) (2007DA10512710408).

References

[1] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Elsevier Computer Networks 47 (4) (2005) 445–487.
[2] N. Alon, A. Bar-Noy, N. Linial, D. Peleg, A lower bound for radio broadcast, Journal of Computer and System Sciences 43 (1991) 290–298.
[3] I. Chlamtac, S. Kutten, On broadcasting in radio networks-problem analysis and protocol design, IEEE Transactions on Communications 33 (1985) 1240–1246.
[4] I. Chlamtac, O. Weinstein, The wave expansion approach to broadcasting in multihop radio networks, IEEE Transactions on Communications 39 (1991) 426–433.
[5] B. Chlebus, D. Kowalski, T. Radzik, On many-to-many communication in packet radio networks, in: 10th International Conference on Principles of Distributed Systems, OPODIS'06, LNCS, vol. 4305, pp. 260–274. Also in Algorithmica, 2009 (in press).
[6] P.G. Doyle, J.L. Snell, Random Walks and Electric Networks, Mathematical Association of America, 1984.
[7] M. Chrobak, L. Gąsieniec, W. Rytter, Fast broadcasting and gossiping in radio networks, Journal of Algorithms 43 (2) (2002) 177–189.
[8] A.E.F. Clementi, A. Monti, R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, in: Proc. 12th Ann. ACM-SIAM Symposium on Discrete Algorithms, SODA'2001, pp. 709–718.
[9] F. Cicalese, F. Manne, Q. Xin, Faster centralized communication in radio networks, in: Proceedings of the 17th International Symposium on Algorithms and Computation ISAAC, in: LNCS, vol. 4288, Springer, 2006, pp. 339–348; The extended version appeared in Algorithmica 54 (2) (2009) 226–242.
[10] A. Dessmark, A. Pelc, Tradeoffs between knowledge and time of communication in gemetric radio networks, in: Proc. of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA 2001, Grete, Greece, July 2001, pp. 59–66.
[11] K. Diks, E. Kranakis, D. Krizanc, A. Pelc, The impact of information on broadcasting time in linear radio networks, Theoretical Computer Science 287 (2002) 449–471.
[12] M. Elkin, G. Kortsarz, Improved broadcast schedule for radio networks, in: Proc. 16th ACM-SIAM Symp. on Discrete Algorithms, 2005, pp. 222–231.
[13] L. Gąsieniec, E. Kranakis, A. Pelc, Q. Xin, Deterministic M2M multicast in radio networks, Theoretical Computer Science 362 (1–3) (2006) 196–206.
[14] I. Gaber, Y. Mansour, Broadcast in radio networks, in: Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms, 1995, pp. 577–585. Also, Journal of Algorithms, 46 (1) (2003) 1–20.

[15] R. Gandhi, S. Parthasarathy, A. Mishra, Minimizing broadcast latency and redundancy in ad hoc networks, in: Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2003, pp. 222–232.
[16] L. Gąsieniec, I. Potapov, Q. Xin, Efficient gossiping in known radio networks, in: Proc. 11th SIROCCO, in: LNCS, vol. 3104, 2004, pp. 173–184; The extended version appeared in Theoretical Computer Science 383 (1) (2007) 45–58.
[17] L. Gąsieniec, D. Peleg, Q. Xin, Faster communication in known topology radio networks, in: Proc. 24th Annual ACM SIGACT-SIGOPS PODC, 2005, pp. 129–137. The extended version appeared in Distributed Computing, 19 (4) (2007) 289–300.
[18] S.C.H. Huang, P.J. Wan, X. Jia, H. Du, W. Shang, Minimum-Latency broadcast scheduling in wireless ad hoc networks, in: Proc. of the 26th Annual IEEE Conference on Computer Communicaitions, IEEE INFOCOM 2007, pp. 733–739.
[19] D. Kowalski, A. Pelc, Optimal deterministic broadcasting in known topology radio networks, Distributed Computing 19 (3) (2007) 185–195.
[20] S.O. Krumke, M.V. Marathe, S.S. Ravi, Models and approximation algorithms for channel assignment in radio networks, Wireless Networks 7 (6) (2001) 575–584.
[21] F. Kuhn, R. Wattenhofer, A. Zollinger, Asymptotically optimal geometric mobile ad-hoc routing, in: Proceedings of the Sixth International Workshop on Discrete Algorithm and Methods for Mobility, 2002, pp. 24–33.
[22] E.L. Lloyd, S. Ramanathan, On the complexity of distance-2 coloring, in: Proc. 4th International Conference on Computing and Information, ICCI'92, pp. 71–74.
[23] F. Manne, S. Wang, Q. Xin, Faster radio broadcast in planar graphs, in: Proc. 4th Conference on Wireless on Demand Network Systems and Services, IEEE press, pp. 9–13. The extended version appeared in Journal of Networks 3 (2) (2008) 9–16.
[24] A. Sen, M.L. Huson, A new model for scheduling packet radio networks, in: Proc. 15th Ann. Joint Conference of the IEEE Computer and Communication Societies, IEEE INFOCOM 1996, pp. 1116–1124.
[25] Q. Xin, F. Manne, Latency-optimal communication in wireless mesh networks, in: Proceedings of the 15th Asia-Pacific Conference on Communications, APCC 2009.
[26] Q. Xin, Y. Zhang, L.T. Yang, Optimal fault-tolerant broadcasting in wireless mesh networks, Wireless Communications and Mobile Computing 11 (5) (2011) 610–620.